

Node.js

Outline

- Web dev overview
- Front-end development
- Back-end technologies
- What is node.js
- Node.js API
- npm, modules & middleware
- Examples

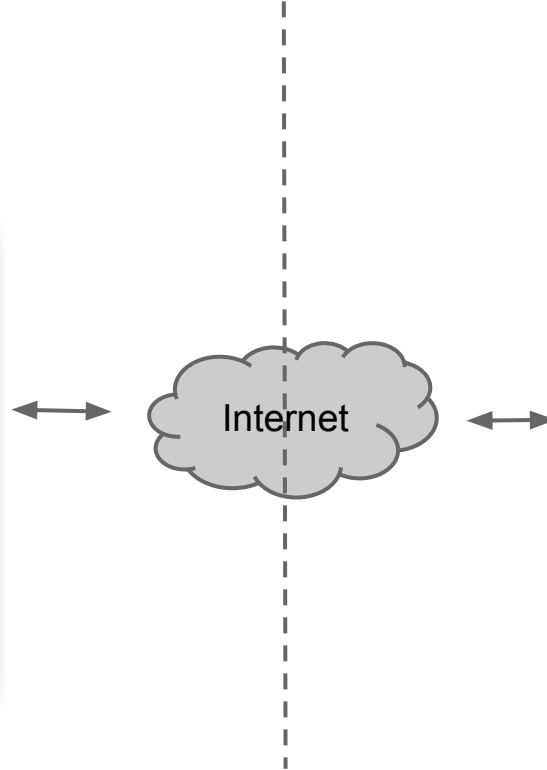
Web development

- The Browser
- Mobile apps
- Third party APIs
- The Cloud

Everywhere!!!

Web development

Front-end



Back-end



HTML - markup

<tag>

<nested>

<element attribute1 attribute2 /> content </element>

</nested>

</tag>

HTML - example

```
<html>
```

```
  <head>
```

```
    <title>Title</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>Hello world</h1>
```

```
  </body>
```

```
</html>
```

// All you need is a browser!

CSS - styling

```
element {  
    style: value;  
}
```

```
body {  
    text-align: center;  
}
```

```
#the_id {  
    color: yellow;  
    background: blue;  
}
```

```
<h1 id="the_id">hello world</h1>
```

```
.a_class {  
    color: #19FF19;  
}
```

CSS - example

```
<style>
  body {
    text-align: center;
  }
  #the_id {
    color: yellow;
    background: blue;
  }
  .a_class {
    color: #19FF19;
  }
</style>
```


Javascript - scripting

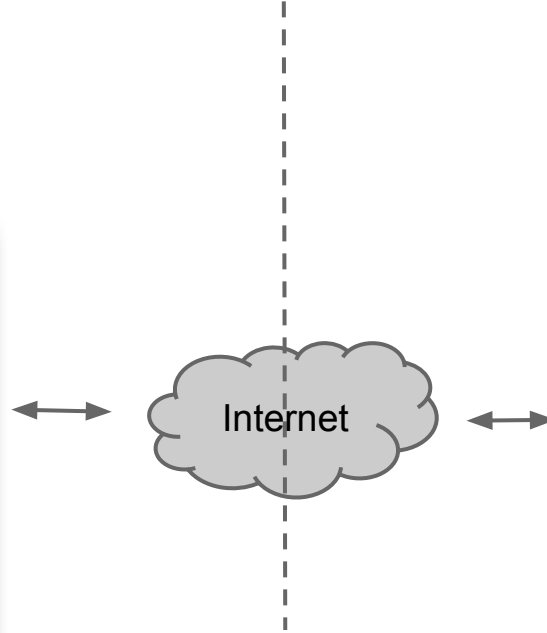
- Programming language
- Javascript + Browser = DOM
 - Access to html elements
 - Massive API
- Javascript + Server = Node.js

Javascript - example

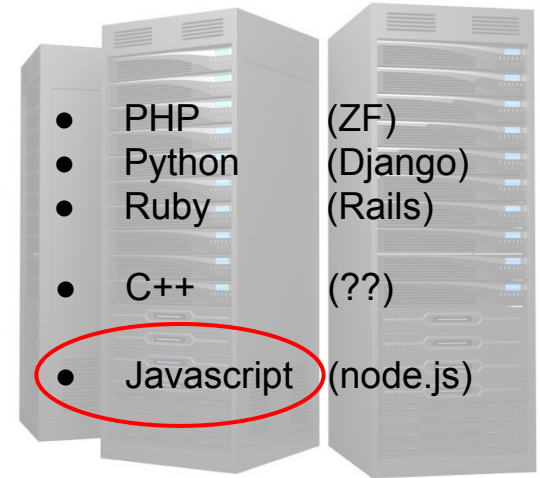
```
<script>
  var clicks = 0
  document.body.onclick = function () {
    alert('You clicked the document '+(clicks++)+' times!')
  }
  var hovers = 0
  document.getElementById('the_id').onmouseover = function () {
    var p = document.createElement('p')
    p.textContent = 'You hovered over the header '+(hovers++)+' times'
    p.className = 'a_class'
    document.body.appendChild(p)
  }
</script>
```

Web development

Front-end



Back-end



Javascript FTW!!

What is node.js?

- Server-side javascript
- Google Chrome's javascript engine
- Repl & command-line javascript
- Not just javascript
 - engine written in c++
 - async I/O (file system & network operations)

Write servers in javascript

Node.js make them **fast** and **efficient**

What is node.js?

See for yourself! <http://nodejs.org/download/>

Node

- an API

npm

- package manager
- huge open-source ecosystem

Node.js API - <http://nodejs.org/api/>

Great add to any developers toolbelt

- File system access
- HTTP
- Sockets
- SSL
- c++ interop

Node.js API - require

```
var module = require('module')  
require('run_a_file.js')
```

Modular code is:

- Readable
- Extensible
- Unit testable

Node.js - example

// from nodejs.org

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(1337, '127.0.0.1');
console.log('Server running at http://127.0.0.1:1337/');
```


Node.js - async

- Function calls placed onto a “call queue”
 - opposite of “call stack”
 - functions will be executed *sometime in the future*



- We write callbacks as “handlers”
- Response to user input or long-running task completion

Node.js - async

```
var http = require('http')           // load the 'http' module from the API

http.createServer(function (req, res) { // callback to function on the request and response

    res.writeHead(200, {'Content-Type': 'text/plain'}) // tell the response header we're sending text

    res.end('Hello World\n')           // send some text

}).listen(1337, '127.0.0.1')          // expose our server to the internet

console.log('Server running at http://127.0.0.1:1337/') // tell us when you're good to go
```

What is node.js?

See for yourself! <http://nodejs.org/download/>

Node

- an API

npm

- package manager
- huge open-source ecosystem

npm - package management

- npm install <module name>
 - access to all prior versions
- huge ecosystem of modules
- npm init
 - creates package.json

npm - package.json

```
{  
  "dependencies": {  
    "express": "latest"  
  }  
}
```

Now we need only run:

```
npm install
```

and our packages will install

A good module

- has an API
- called / instantiated with `require('module')`
- makes your life easier

Old, confusing example

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(1337, '127.0.0.1');
console.log('Server running at http://127.0.0.1:1337/');
```

New and improved example

```
var app = require('express')()
```

```
app.get('/', function (req, res) {  
  res.send('Hello world')  
})
```

```
app.listen(3000)
```